NOTE: Use these buttons to print or save the form. DO NOT use the browser tool bar.

SAVE PRINT

LA-UR-09-06025

Approved for public release; distribution is unlimited.

Title:

Increasing Model Efficiency for High-Resolution Baron Fork Simulations Using Basin Structure Characteristics

Author(s):

Susan Mniszewski, CCS-3
Patricia Fasel, CCS-3
Enrique Vivoni, Arizona State University
Amanda White, EES-14
Everett Springer, STBPO-PRM

Intended for:

9th Annual SAHRA Meeting
September 23-24, 2009
Tucson, AZ

Los Alamos NATIONAL LABORATORY EST. 1943

Increased Efficiency for High-Resolution Baron Fork Simulations Using Basin Structure Characteristics

Sue Mniszewski, Pat Fasel, Enrique Vivoni, Amanda White, Everett Springer

LAUR-09-06025

Abstract

The growing trend of model complexity, data availability and physical representation for watershed simulations has not been matched by adequate developments in computational efficiency. This situation has created a serious bottleneck that limits existing hydrologic models to small domains and short durations. A novel parallel approach has been applied to the TIN-based Real-Time Integrated Basin Simulator (tRIBS), which provides continuous hydrologic simulation using a multiple resolution representation of complex terrain based on a triangulated irregular network (TIN). Our approach utilizes domain decomposition based on sub-basins of a watershed. A stream reach graph based on the channel network structure is used to determine each sub-basin and its connectivity. Individual sub-basins or sub-graphs of sub-basins are assigned to separate processors to carry out internal hydrologic computations (e.g. rainfall-runoff transformation). Routed streamflow from each sub-basin forms the major hydrologic data exchange along the stream reach graph. Individual sub-basins also share subsurface hydrologic fluxes across adjacent boundaries. A timesaving capability known as MeshBuilder has been developed to allow the unstructured mesh and stream flow network for very large basin experiments to be created only once, where multiple runs are required. A tRIBSReader Visualizer (based on ParaView) provides model debugging and results presentation. In the context of a high-resolution Baron Fork basin model (~900K nodes), multi-constraint graph partitioning based on node count, stream reach network connectivity and subsurface flux network connectivity is shown to increase scalability and performance significantly.

Increased Efficiency for High-Resolution Baron Fork Simulations Using Basin Structure Characteristics

9th SAHRA Annual Meeting
September 23-24, 2009

Sue Mniszewski, Pat Fasel, Enrique Vivoni,
Amanda White, Everett Springer

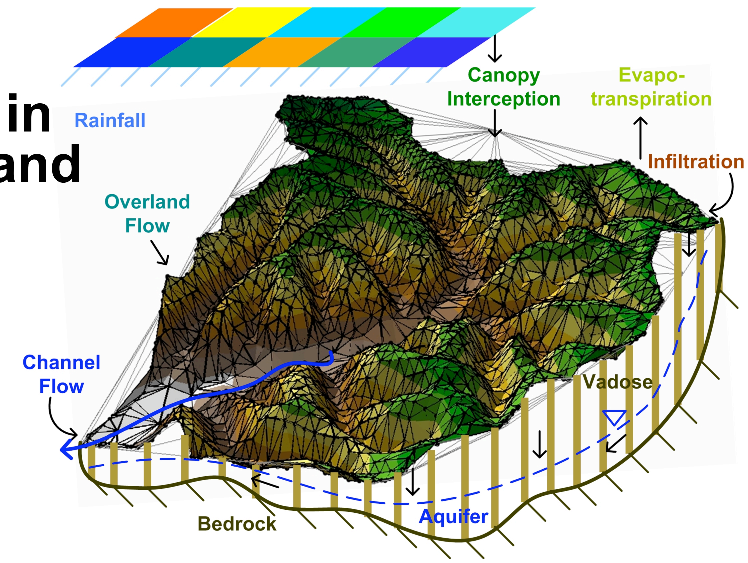
LAUR-09-06025

Introduction

- **Watershed simulations are increasing in model complexity, data availability, and physical representation**
- **New tools and methods are required to improve computational efficiency**
- **Contributions in the context of the tRIBS Simulator and a high-resolution Baron Fork basin model include code parallelization, mesh preprocessing, visualization, and structure-based partitioning**

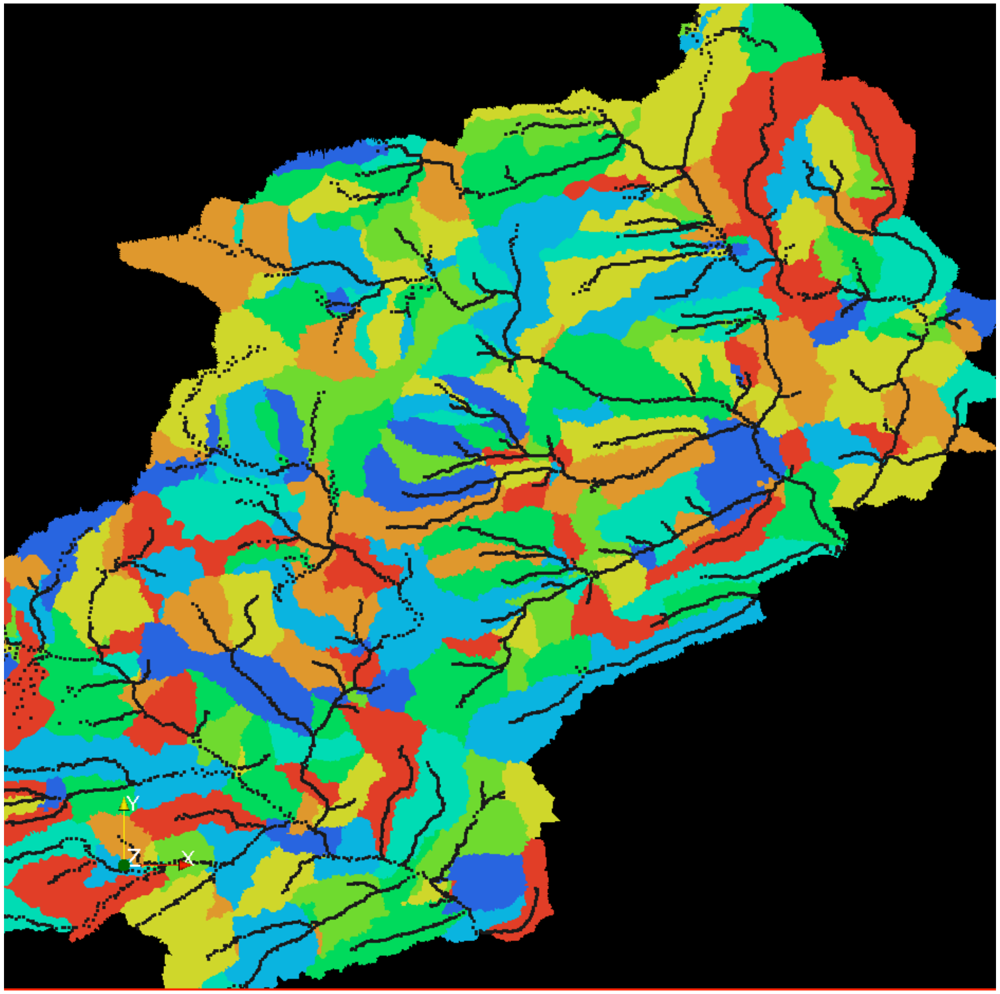
Parallel TIN-based Real-Time Integrated Basin Simulator (tRIBS)

- Collaboration with Enrique Vivoni (Arizona State University)
- Physically-based, 1-D stream, 2-D surface, 3-D sub-surface
- Accounts for rainfall interception, evapotransporation, moisture dynamics in the unsaturated and unsaturated zones and runoff routing
- Collection of C++ classes for distributed hydrological modeling
- Creates DEM-based mesh and stream network
- Domain decomposition based on sub-basins of a watershed



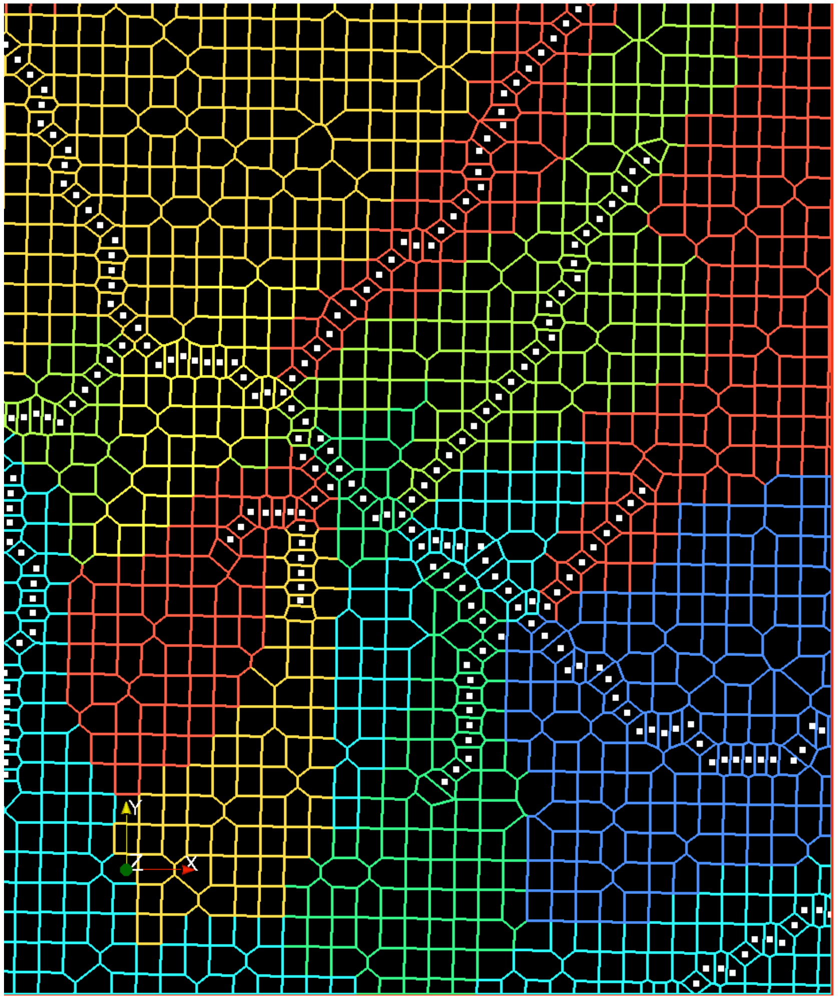
Sub-basin Decomposition

- **Channel network structure determines each sub-basin and its connectivity**
- **A sub-basin consists of a stream reach and contributing area**



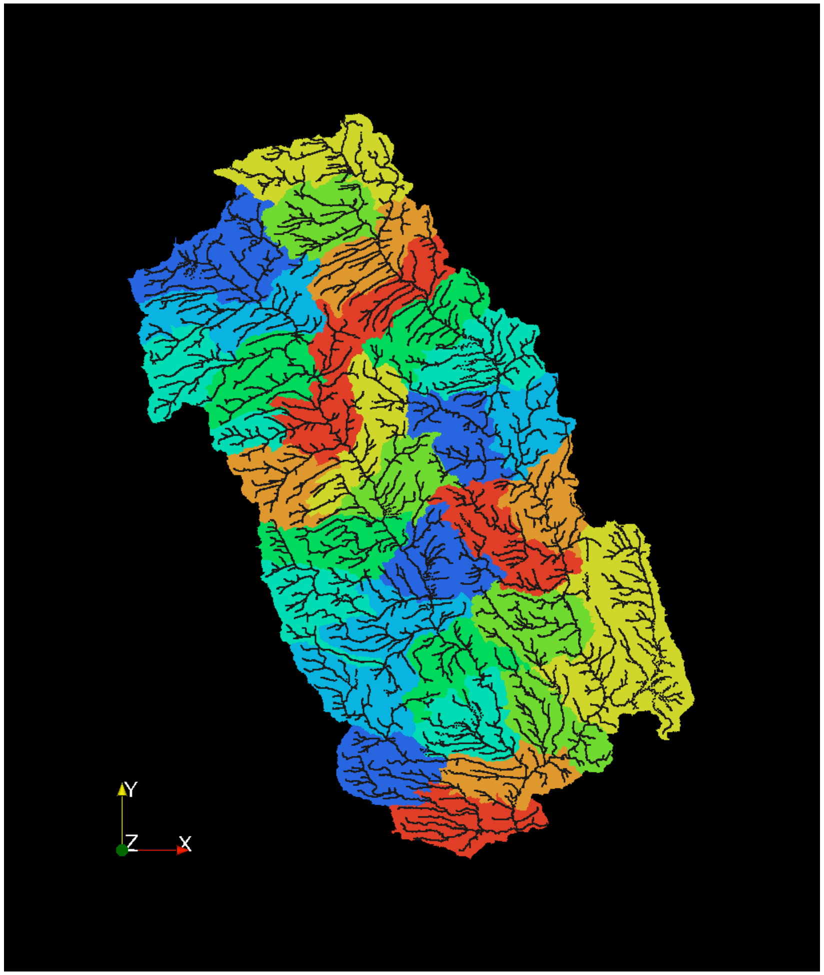
Reaches in Detail

- Composed of voronoi polygons (nodes) from stream and area contributions
- A node is the smallest computational element
- Node counts can vary across reaches



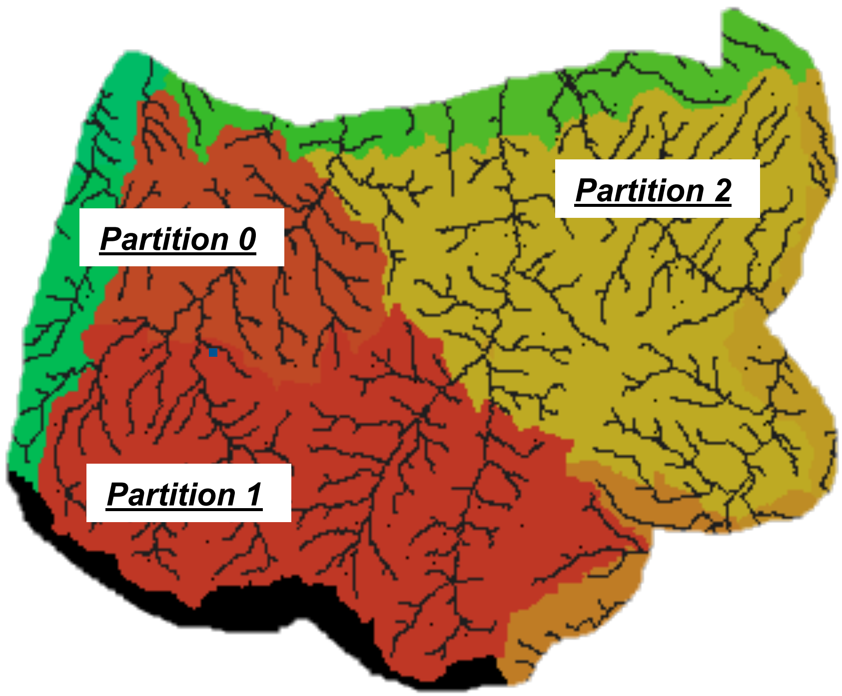
Sub-basin Distribution

- **Individual sub-basins or sub-graphs of sub-basins compose partitions assigned to separate processors for hydrological computation**
- **Data exchanges between processors include**
  - Routed streamflow along the stream reach graph
  - Subsurface hydrologic fluxes across adjacent boundaries



Ghost Cells

- **Ghost cells are required at boundaries between partitions to hold relevant state information for exchange**
- **Surface – Unsaturated Zone**
  - Reach: outlet -> downstream head
    - ***Unsaturated lateral flow***
  - Reach: head -> upstream outlet
    - ***Discharge***
    - ***Depth to groundwater table***
    - ***Wetting front depth***
- **Subsurface – Saturated Zone**
  - Flux: local -> remote
    - ***Depth to groundwater table***
  - Flux: remote -> local
    - ***Groundwater change***
- **Stream – River Routing**
  - Reach: outlet -> downstream head
    - ***Discharge***



Building Mesh and Stream Flow Network - MeshBuilder

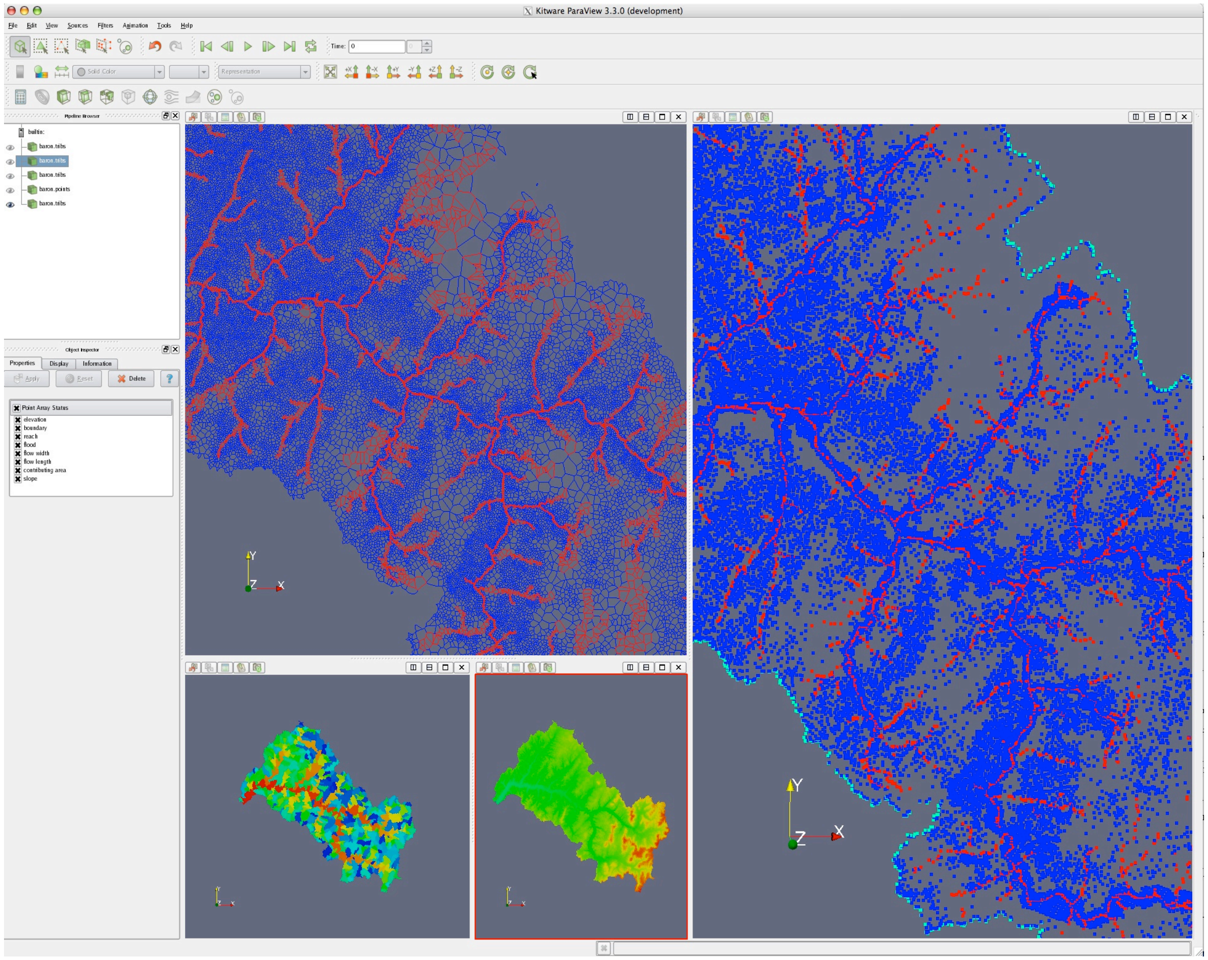
- **Allows large problems and faster startup**
- **Mesh and stream flow network created once for multiple runs**
- **Process similar to tRIBS – streamlined**
  - Runs serially
  - Preprocesses point file to create mesh by reach
  - Produces voronoi nodes, edges, flow, reach, and flux information
  - Includes ghost cell lists required per reach
- **Parallel tRIBS "Option 9" runs**
  - Makes tRIBS data parallel
  - Each processor reads only its assigned set of reaches
  - Different partitioning schemes can be specified
- **Successfully runs on ~900K node Baron Fork basin, 3.6M node Rio Grande basin**

Restart Capability

- **Necessary for long running simulations**
- **Useful for varying run scenarios after an initial time period**
- **Variables are dumped in binary format, one file per processor**
  - Runs must continue on the same number of processors
- **User specifies restart interval, directory, and mode**
- **Restart files can be post-processed for anomaly detection and statistical analysis**

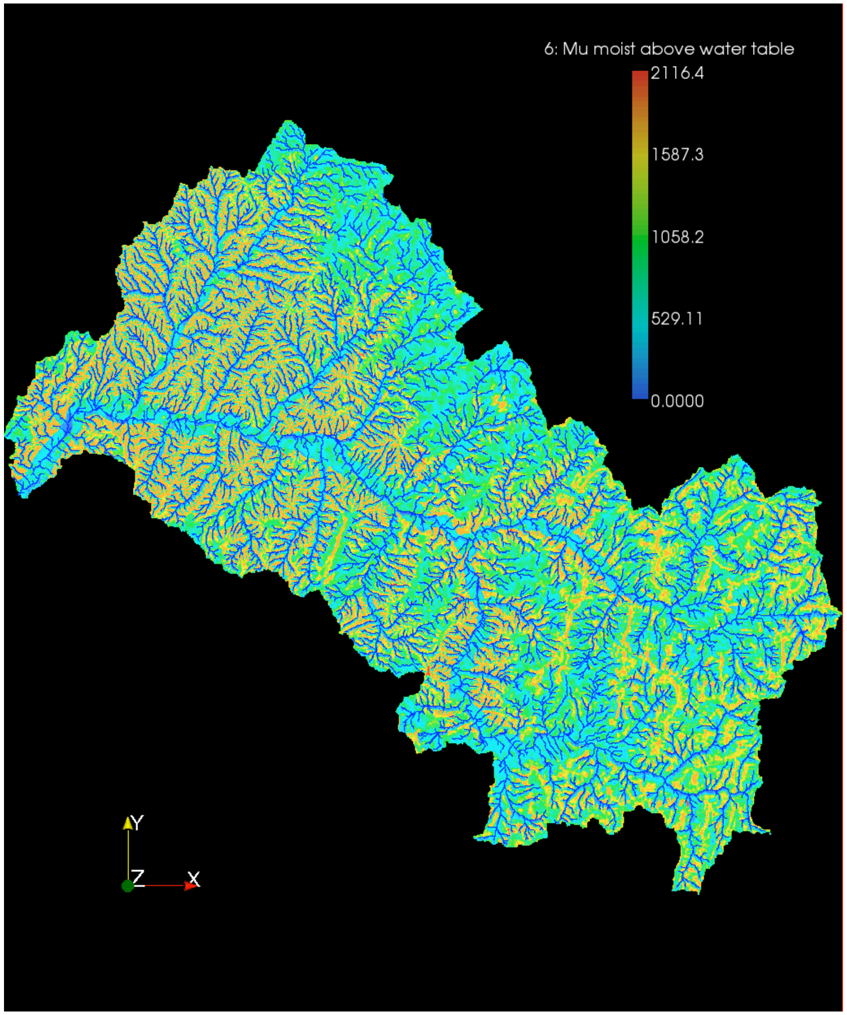
tRIBSReader Visualizer

- ParaView (OpenSource) or Ensight plugin
- Handles very large basins
- Useful for debugging points file, mesh, flow network, partitioning, and simulation
- Collect binary data written per cell for each output interval in tRIBS
- View unstructured grids of polygons where cells are colored by static or dynamic variables (ex. elevation, soil moisture)



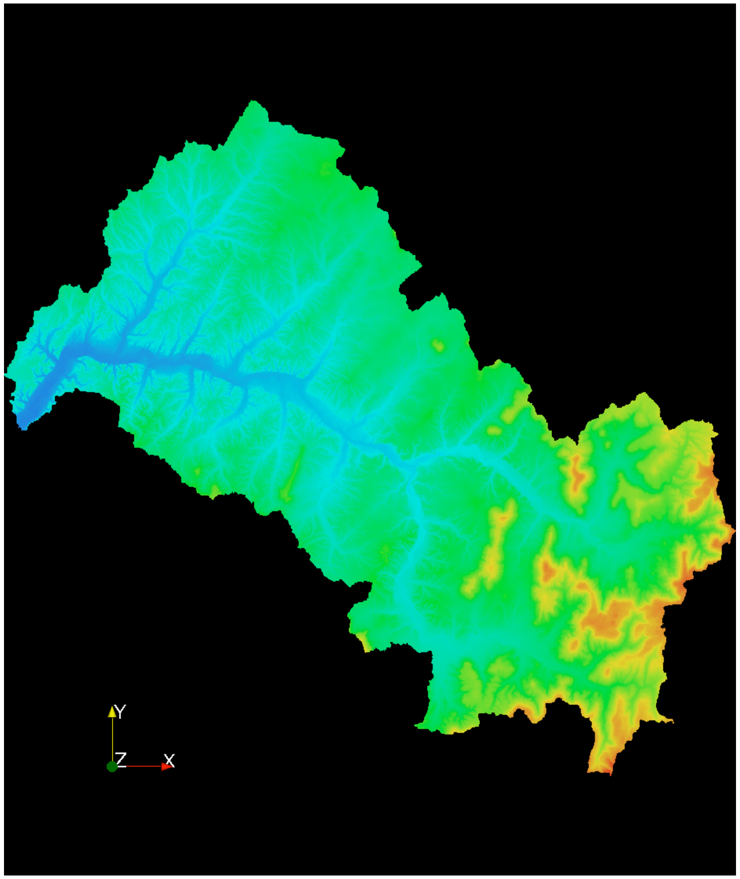
Viewing Time Series Data

- **View dynamic data per time step for debugging and results presentation**



High-Resolution Baron Fork Basin (OK)

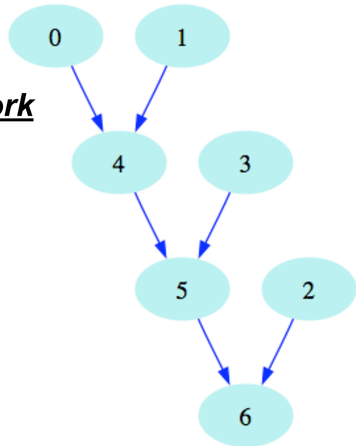
- **~900K nodes, ~5M edges, 5707 reaches**
- **Run Nov 1, 1997 – Dec 1, 1998**
- **On 1 processor**
  - Run time = 15 hours for ~10 days
  - 92.479 min/simulated day
- **Determine efficient performance using first 30 days**
  - Run on 32, 64, 128, and 256 processors
  - How many processors are required?
  - What is the best partitioning of reaches across processors?

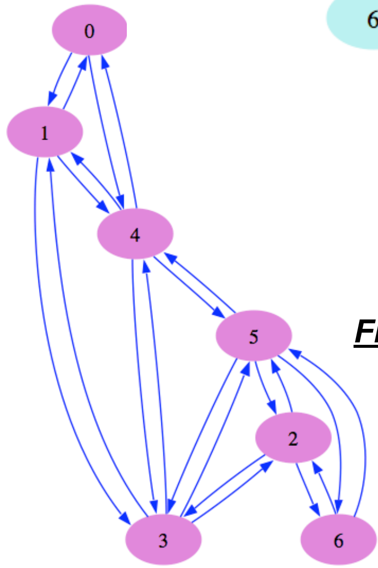


Partitioning

- Balance computation and message passing per processor
- Number of nodes per reach contributes to computational load
- Connections between reaches in the stream network and subsurface flux network contribute to messaging
- Using Metis for multi-constraint graph partitioning

Flow Network

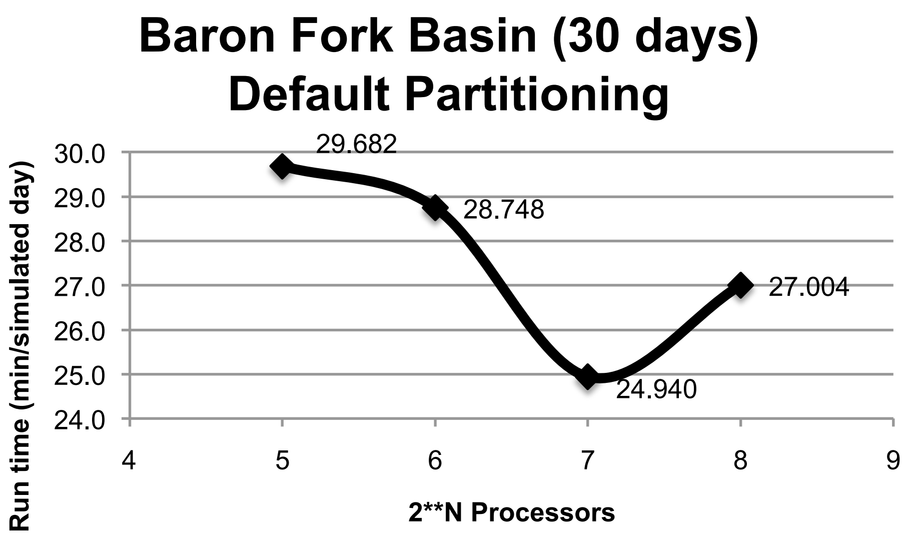


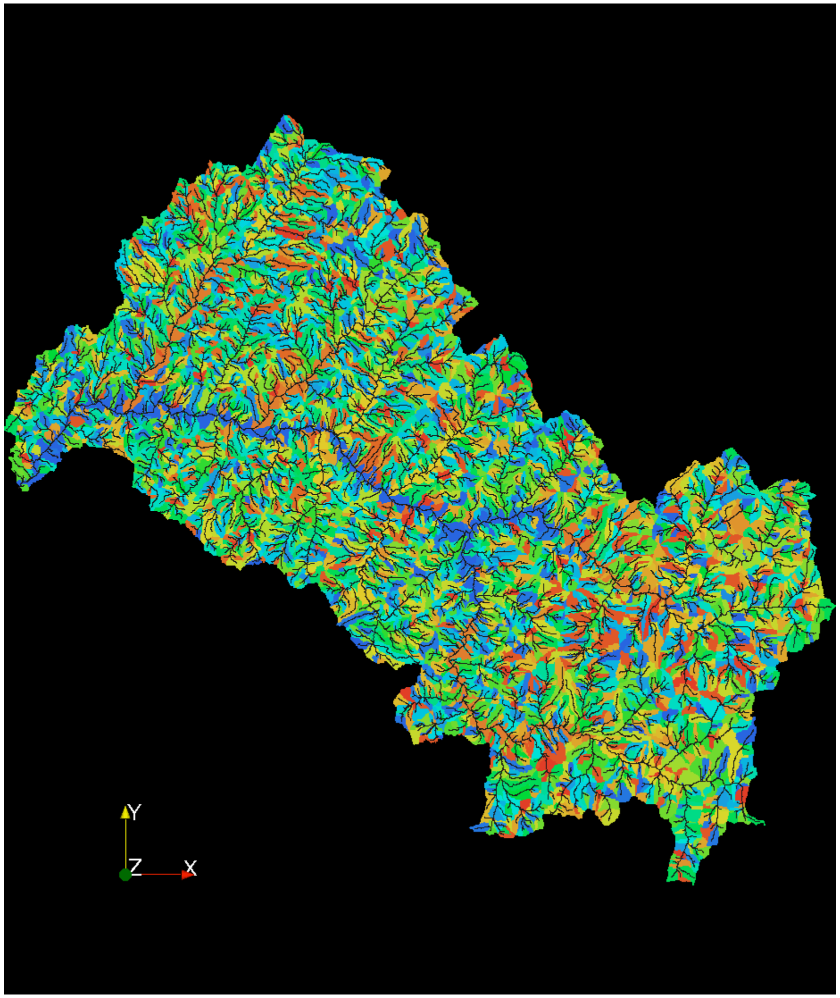


Flux Network

tRIBS Default Partitioning

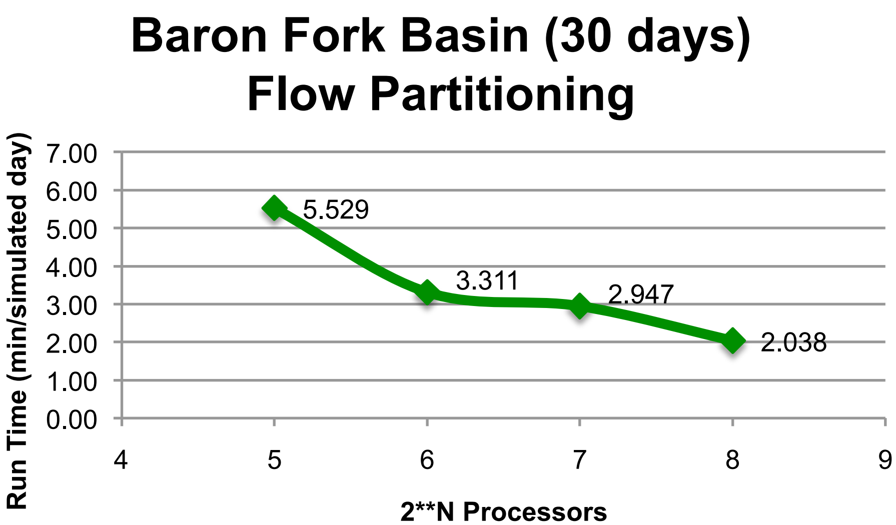
- Based on order that reaches are derived

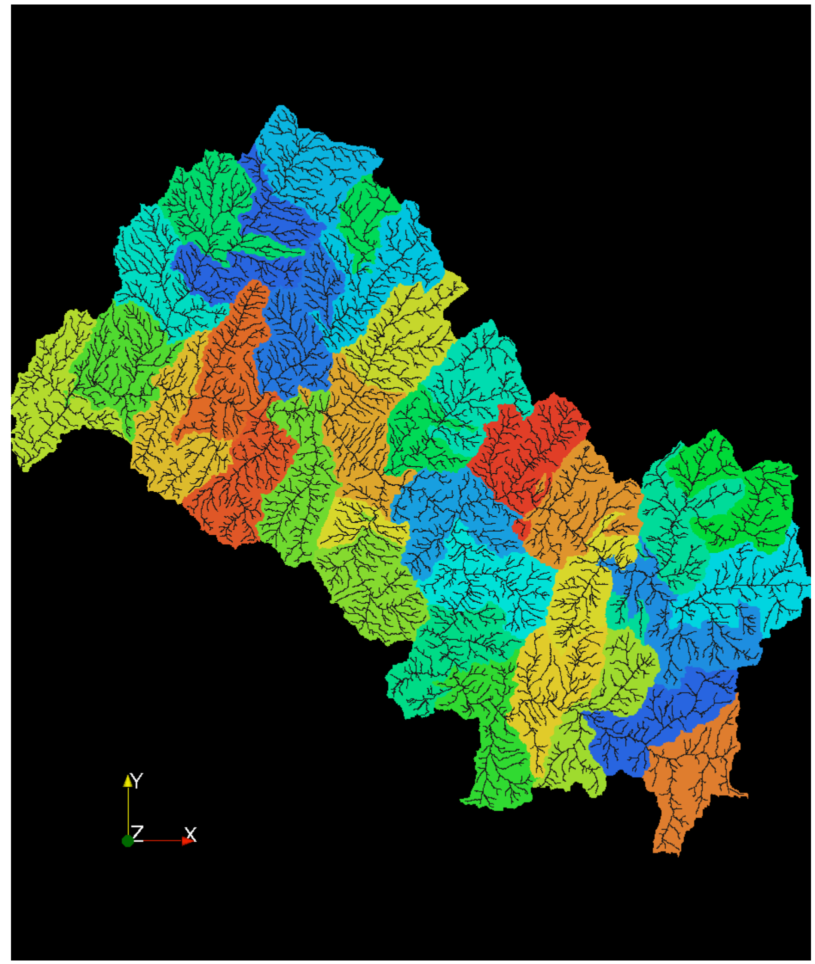




Flow Partitioning

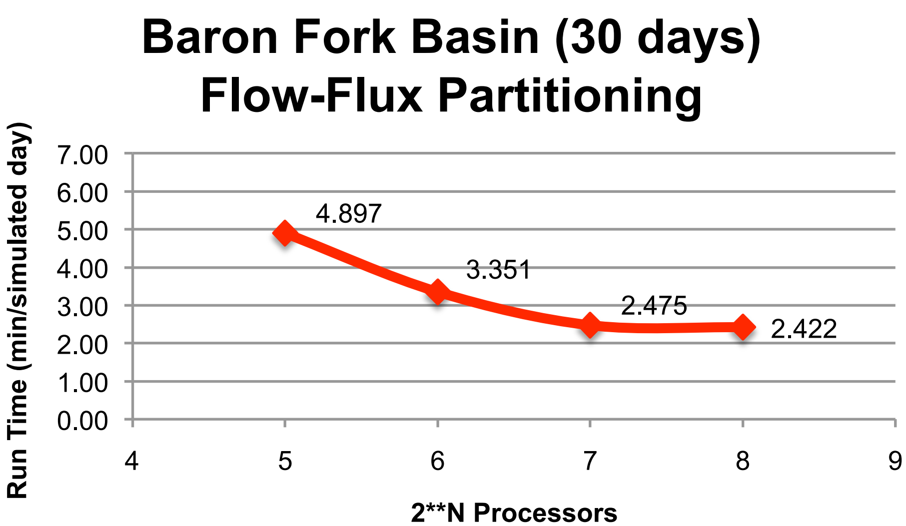
- **Balance number of nodes and stream reach network connectivity**

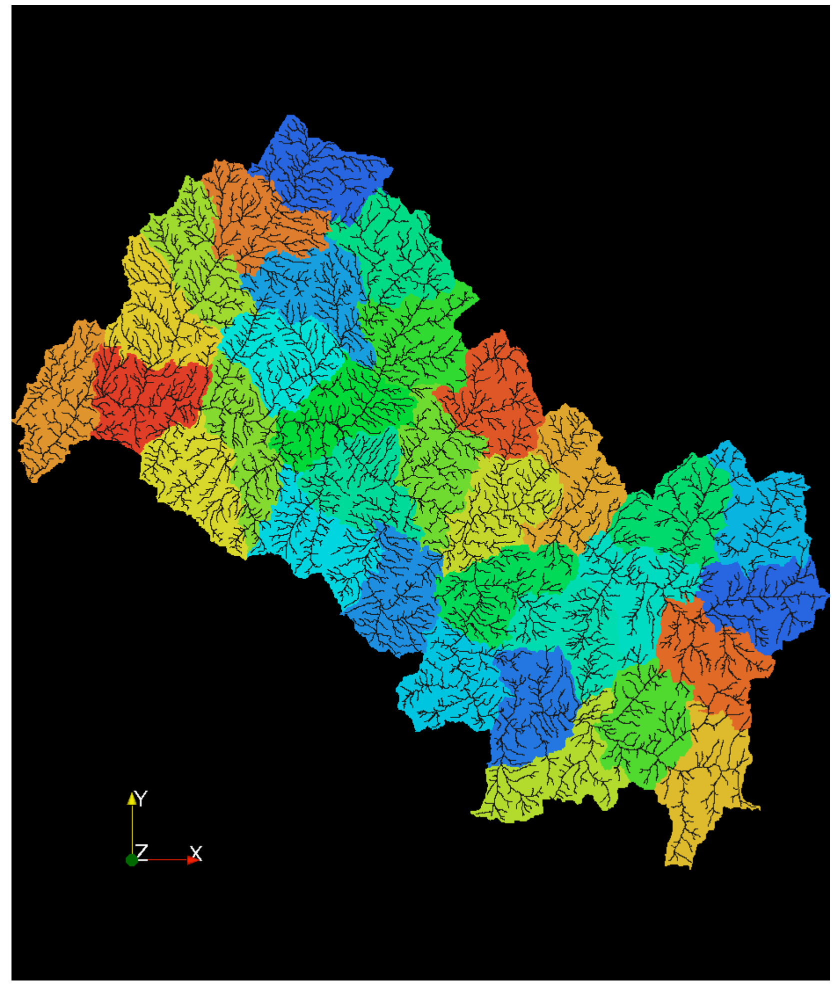




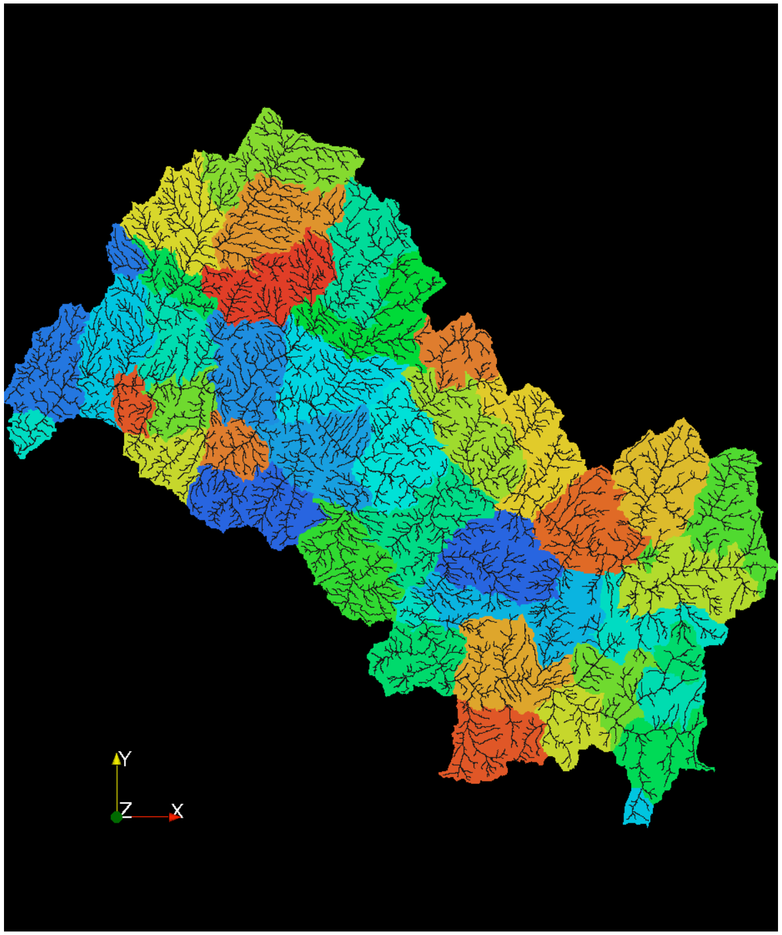
Flow-Flux Partitioning

- **Balance number of nodes, stream reach network and subsurface flux connectivity**

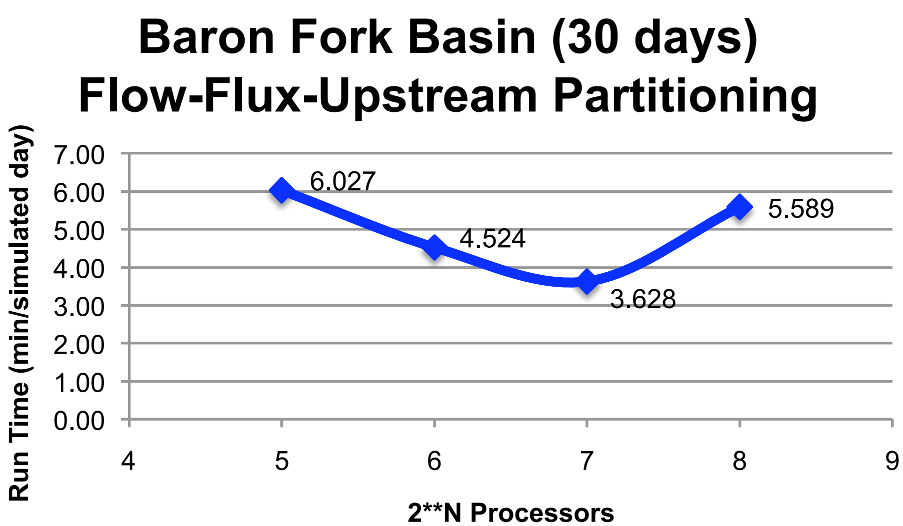




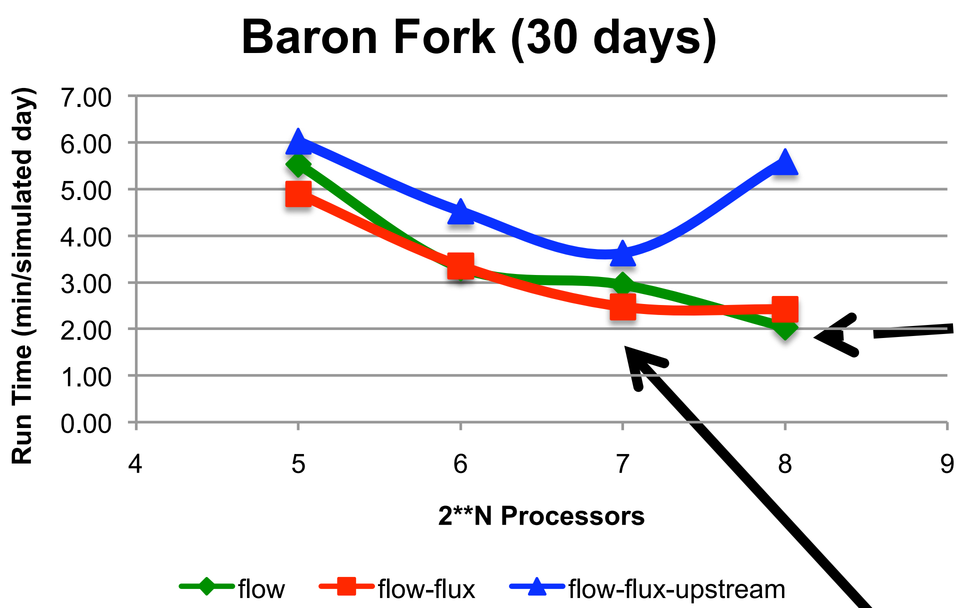
Flow-Flux-Upstream Partitioning



- **Balance number of nodes, stream reach network and subsurface flux connectivity, and number of reaches without upstream reaches**



How many processors?



256 processors!!

For Flow

128 processors!!

For Flow-Flux & Flow-Flux-Upstream

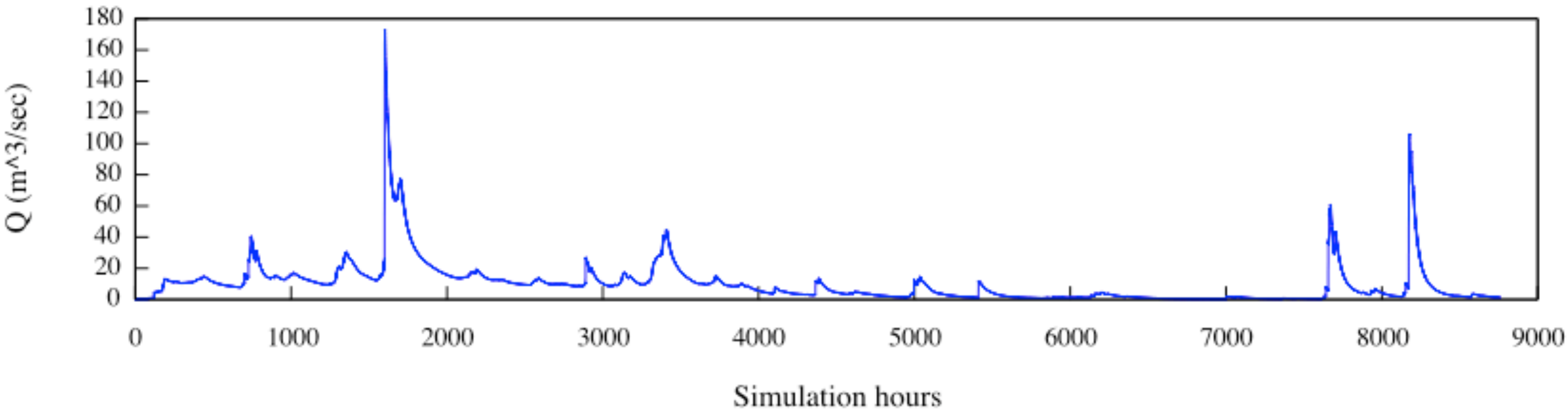
Best Partitioning?

Partition Averages and Standard Deviations for 32 processors

	Default	Flow	Flow-Flux	Flow-Flux-Upstream
# Nodes	27,193.41 6511.65	27,193.41 932.76	27,193.41 516.32	27,193.41 922.20
# Reaches	178.34 0.48	178.34 24.55	178.34 20.86	178.34 7.70
# Downstream Partitions	11.75 5.44	1.41 0.76	2.75 1.32	3.5 1.34
# Flux Partitions	31.00 0.00	4.69 1.91	4.44 1.41	5.5 1.72

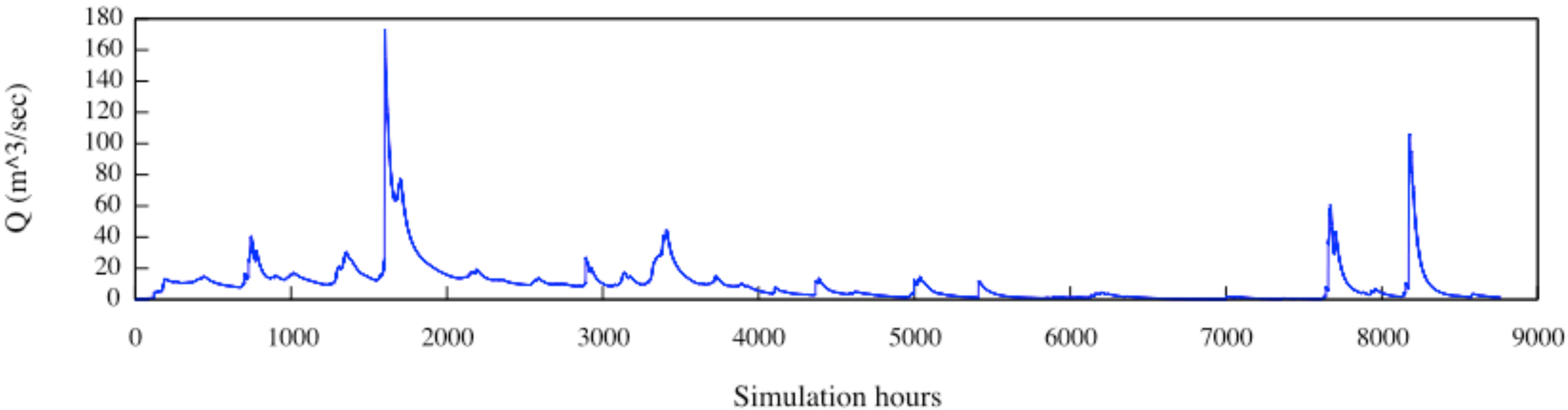
Flow-Flux provides the best balance of nodes and flux connections.

Baron Fork 1 Year Runs - 1997-1998 (128 processors)



Partitioning	Total Run Time (hr)	Run Time (min/simulated day)	Speedup	Efficiency
Flow	> 15.0	2.504	36.932	0.289
Flow-Flux	13.3	2.186	42.305	0.331
Flow-Flux-Upstream	> 15.0	3.400	27.200	0.212

Baron Fork 1 Year Runs - 1997-1998 (256 processors)



Partitioning	Total Run Time (hr)	Run Time (min/simulated day)	Speedup	Efficiency
Flow	9.9	1.623	56.980	0.223
Flow-Flux	12.1	2.000	46.239	0.181
Flow-Flux-Upstream	> 15.0	4.419	20.927	0.082

Summary

- tRIBS has been extended to run in parallel for larger basin models
- Creating the mesh and stream flow network only once, using MeshBuilder, saves simulation startup time
- Visualization using tRIBSReader is useful for model debugging and results presentation
- Balancing of node counts and connectivity of the stream reach and subsurface flux networks per processor produces faster running simulations
- Computational efficiency demonstrated implies the feasibility of larger and higher-resolution watershed simulations